Dossier PLSQL

set serveroutput on;

--Question n°1--------

-- Créer une procédure pour inscrire un adhérent à une activité. Ajouter le nombre d'invités conviés.

--L'inscription sera refusée si la date limite d'inscription est dépassée, ou s'il y a plus de 50

--participants à l'activité

create or replace procedure inscrire\_adh\_activite

(

v\_id\_adh in membre.id\_membre%type,

v\_id\_act in activite.id\_activite%type,

v\_nb\_inv in participer.nombre\_invites%type

)

as

v\_date\_activite date;

v\_date\_du\_jour date;

date\_limite\_inscription date;

v\_nombre\_de\_participants number(10):=0;

begin

select date\_activite into v\_date\_activite from activite where id\_activite = v\_id\_act;

SELECT SYSDATE INTO v\_date\_du\_jour FROM dual;

select count(id\_membre)+sum(nombre\_invites) into v\_nombre\_de\_participants

from participer

where id\_activite = v\_id\_act;

date\_limite\_inscription := v\_date\_activite -5;

dbms\_output.put\_line('v\_date\_activite '|| v\_date\_activite);

dbms\_output.put\_line('date\_limite\_inscription '|| date\_limite\_inscription);

dbms\_output.put\_line('v\_nombre\_de\_participants ' || v\_nombre\_de\_participants);

if((v\_date\_du\_jour > date\_limite\_inscription)) then

dbms\_output.put\_line('la date de l''activite est trop rapprochée, ou l''activité est déjà passée');

RAISE\_APPLICATION\_ERROR (-20001, 'Date de l''activité trop rapprochée ou dépassée');

end if;

if (v\_nombre\_de\_participants>50) then

dbms\_output.put\_line('Trop de monde dans l''activité');

RAISE\_APPLICATION\_ERROR(-20002, 'le nombre de participants est supérieur à 50');

end if;

--dbms\_output.put\_line('On va inscrire l''adhérent à l''activité');

--dbms\_output.put\_line('Mais on va d''abord tester qu''il n''est pas déjà inscrit à cette même activité');

if(verifinscription(v\_id\_adh,v\_id\_act)) then

--dbms\_output.put\_line('l''adhérent n''est pas déjà inscrit à cette activité, on va donc l''y inscrire');

insert into participer (id\_membre, id\_activite, date\_inscription, nombre\_invites) values (v\_id\_adh,v\_id\_act,v\_date\_du\_jour, v\_nb\_inv);

--mais il faudrait également connaître le nombre d'invités et leurs noms, que je vais ignorer ici.

else

RAISE\_APPLICATION\_ERROR(-20002, 'l''adhérent est déjà inscrit');

end if;

end inscrire\_adh\_activite;

---------------------

-- Vérifier qu'un adhérent n'est pas déjà inscrit à une activité

create or replace function verifinscription

(

v\_id\_adh in membre.id\_membre%type,

v\_id\_act in activite.id\_activite%type

)

return boolean

as

v\_adherent\_estinscrit number(1);

begin

--select count(\*) into v\_adherent\_existe from membre where id\_membre = v\_id\_adh;

select count(\*) into v\_adherent\_estinscrit from participer where id\_membre = v\_id\_adh and id\_activite = v\_id\_act;

if(v\_adherent\_estinscrit =0)then

return true;

else return false;

end if;

end verifinscription;

-----lancer l'inscription avec pour paramètres l'id adhérent, l'id de l'activité et le nombre de

-----participants

begin

inscrire\_adh\_activite(3,8,3);

--dbms\_output.put\_line('Est-ce que untel est inscrit? ' || verifinscription(1,10));

--Question n°2----------------

--Créer une fonction pour valider une adresse e-mail